REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
-7/23/99

3. REPORT TYPE AND DATES COVERED
Final, 12/1/95-11/30/97

4. TITLE AND SUBTITLE
DOMINION: System Structure, Applications, and Commercial Viability

5. FUNDING NUMBERS
DAAH04-96-C-0015

6. AUTHOR(S)
Donald L. Miller

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Advanced Process Combinatorics, Inc. P.O. Box 2324, West Lafayette, IN 47996-2324

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

ARO 35052.1-RT-ST2

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by the documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The project involved the development of specialized algorithmic methods for sequential and distributed computing environments aimed at the solution of large-scale design and production scheduling problems in the chemical and pharmaceutical industries. This project has been highly successful as evidenced by the level of new technology developed, the degree to which this technology has been successfully applied to problems in the target area, and the commercial success which has been achieved, measured in sales revenues of software products that utilize this technology.

14. SUBJECT TERMS
Distributed Computing, Optimization, Scheduling, Design, Planning

15. NUMBER OF PAGES
13

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500 **Standard Form 298**
(Rev.2-89)
Prescribed by ANSI Std. 239-18
298-102

DOMINION:
System Structure, Applications, and Commercial Viability

Final Report for Contract DAAH04-96-C-0015

The project involved the development of specialized algorithmic methods for sequential and distributed computing environments aimed at the solution of large-scale design and production scheduling problems in the chemical and pharmaceutical industries. This project has been highly successful as evidenced by the level of new technology developed, the degree to which this technology has been successfully applied to problems in the target area, and the commercial success which has been achieved, measured in sales revenues of software products that utilize this technology.

Process Design and Parallel/Distributed Computing

A key result of this work has been new technology for the use of sequential, parallel and distributed computing environments to address the complex problem of batch chemical process design. Batch processes were chosen as a target area because of their prevalence in the manufacture of high value added products such as specialty chemicals and pharmaceuticals. The highly competitive nature of these businesses has created a need for more efficient process designs and hence for a methodology for developing these designs rapidly. The developed approach involves modeling the design problem as a mathematical program and solving the problem using optimization techniques. The goal is to maximize or minimize an objective function for costs or profits, subject to physical and operational limits, handled as constraints in the mathematical program. The technology developed during this project has greatly increased the size and complexity of chemical and pharmaceutical process design and scheduling problems that may be reliably solved. This progress has been achieved through research in two areas, first through the development of specialized algorithms that exploit the special structure of these problems, and second through the use of distributed computing environments.

In this project, two distinct types of problems faced in process design were targeted: process configuration and process scheduling, for optimizing high level discrete decisions about how a process should be built (the former) and lower level detailed verification of the operability of individual configurations (the latter). A modeling language for each of these applications was developed, DSPEC for configuration and RCSPEC for scheduling. These languages provide a natural description of process configuration and scheduling problems. Problems are automatically translated into rigorous mathematical programs, specifically Mixed Integer Linear Programs (MILPs). These MILPs have proved difficult to solve using general-purpose software. This difficulty has been mitigated through the development of special purpose solution technology. The solver engines developed during this project take advantage of the underlying structure of this class of problems, allowing routine solution of real world process design problems to within less than 1% of the best possible solution, often using less than an hour of computer time.

A solution to the configuration portion of process design problems specifies the type and number of particular equipment items that should be purchased as well as the timing of plant expansions. Consider for example, a new product for which the sales forecast grows rapidly during the first five years of plant operation. An optimal configuration may involve building an initial plant coupled with strategic expansions as sales increase. The equations involved in the model of the configuration problem treat the time required to run the tasks on various pieces of process equipment, precedence constraints with regard to necessary feed stocks being produced before tasks are started and the overall availability of resources such as electricity and labor. These equations do not however, treat the details of generating a schedule on an individual task by task basis as this would overcomplicate the solution of a higher level decision problem with details that are not necessary until a small group of sensible candidate configurations are determined for further examination. Thus the resulting configuration is not guaranteed to have a true capacity equal to that required for effectual detailed scheduling. All of these details are considered in the subsequent solution of detailed scheduling problems.

The design process also involves decision making in the light of uncertainty in the predictions of the market demand. In the past, stochastic parameters have been addressed by representing product demands as continuous variables within certain bounds (which are determined *a priori*. Uncertainty was incorporated here in the form of probability distributions through the development of a method to utilize the concept of scenarios, explicitly handling uncertainty within a manageable linear formulation. Scenarios are a collection of predicted demand levels along with their associated probabilities. The objective of the design is to find an optimal configuration and operable production schedule that takes into account all the scenarios in the decision making.

A formal description of the process design problem contains:

- The process recipe structure (Multi-level Bills of Materials)
- Product demand patterns (with or without uncertainty)
- Process data (yields, processing time, etc.)
- Scheduling constraints (change-over, resource availability, etc.)
- Cost data for equipment purchase and utilization
- The horizon of interest (life of the plant)
- Cost data for inventory and utilities (steam, manpower, etc.)
- The configuration of equipment over time (unit purchase over time)
- Production plan over time (what and when to produce)
- Detailed schedules to follow the production plans
- Depreciation and tax rates which allow determination of optimal return on investment

The natural decomposition of the combined configuration-scheduling model was exploited based on the differing time scales involved, one based on the aggregate planning (macroscopic) features and the other, based on the scheduling (microscopic) details. This separation makes solution of the underlying problems more logical and tractable. Separation of the design problem into two stages still implies interactions between the stages, and one cannot be solved without taking into account the influence of

the other. The developed approach considers configuration and scheduling as a two-tier problem with an interactive interface between the two stages. The interface serves to link the two levels and keep them mutually consistent. The solution of the configuration problem at the top level, called the Design SuperProblem (DSP), is used by the interface to set up a series of scheduling problems (Scheduling SubStage) taking into account boundary conditions implied by the DSP solution. The DSP solution therefore serves as a target that the scheduling problems should meet to achieve the desired goals. Using this approach, design solutions of good quality were developed, without compromising important aspects of the physics of the problem. This is accomplished using an iterative decomposition procedure. In this procedure, it is sometimes necessary to include additional equipment capacity to satisfy scheduling constraints.

The MILPs resulting from the DSP may be solved using branch and bound techniques that map naturally onto parallel and distributed computers. Although this algorithm class contains a large degree of inherent parallelism, traditionally parallel computing environments have required special purpose programming which was not portable to other architectures or operating systems. This, coupled with the rapid pace of microprocessor improvement, yields a paradoxical situation in which a parallel computer with two year old processors is slower than this year's personal computer. Often the time involved to port a program to a parallel computer was longer than the period of time for which that computer may be expected to provide state-of-the-art performance. This explains much of the lack of parallel computer technology in commercial use today. While the power of parallel computing presented an attractive avenue for use in solving large MILPs involved in process design optimization, we wished to avoid the problem of porting to various parallel computing environments which could prove to be dead ends. Fortunately, this pitfall was avoidable due to advanced communication environments developed elsewhere.

A system called Parallel Virtual Machine (PVM) was developed at the Oak Ridge National Laboratory. This public domain system provides a flexible and portable framework on which to build parallel and distributed computer software applications. Because of its support for many different computer architectures, PVM was used as the foundation for the parallel/distributed MILP solver, named Dominion, developed in this project. The choice of PVM for a platform relieved APC of the need to customize code the low level communication protocols needed for inter-process communication. PVM provides the basic ability to manage a virtual cluster of machines of various architectures ranging from personal computers to Cray supercomputers. Machines may be added to the cluster, dropped from it, on an interactive basis using calls to PVM library routines. This, along with the choice of C++ as a development language, provided a built-in portability of the Dominion solver across a wide range of architectures.

With the original parallel MILP implementation used in phase I of this project as a base, the new generation of Dominion was developed using the PVM system. The overall computational model for distributed MILP solution proved to be both effective and robust. Results on standard MILPs from the literature have been encouraging, as Dominion is routinely able to use up to a dozen networked computers to solve MILPs 8-

10 times faster than a single machine. This performance has been equaled or surpassed on the specific MILPs that occur in the DSP stage of process design applications. This is because the DSP problems are addressed using customized heuristic methods that operate on nodes in the branch and bound tree. Problems that are highly constrained may require the solution of several thousand nodes, with each of these nodes contributing to the probability of the heuristic finding a good feasible solution.

The computational model developed for Dominion is based on a master process that directs a collection of worker processes that evaluate the nodes in the branch and bound search tree. The master keeps track of the number of unexplored nodes available to send out to new workers, the best feasible solution reported by the workers and the number of worker processes spawned. The master also serves an important function in determining when all work is completed and terminating the workers. New worker processes are not created unless there is sufficient work to support them. Worker processes retain their own local work queue and draw new problems from that queue preferentially. This minimizes the amount of network traffic and reduces latency. Workers report improvements in the feasible solution immediately upon discovery. They also periodically broadcast the size of their local work queue and the quality of the best lower bounded node in that queue. Workers keep track of these broadcasts received from other worker processes so that when more work is needed, they can contact the worker that has the most attractive nodes in its local queue. When workers receive requests for nodes they respond by sending nodes from their local queue to the requesting process. Performance statistics collected on real-world process design problems indicate that this system places almost no load on the network. Typically, several hundred nodes are solved for every node shipped across the network and processors spend very little time outside of their major activity of solving branch and bound nodes and running the heuristics for seeking feasible solutions.

Algorithm Engineering

While our model for distributed computing provides performance enhancements, the value to be had here is on the order of a factor of ten to one hundred. This can indeed prove useful, but only as a method of speeding up an already well designed algorithm. The nature of these design and scheduling problems is such that any attempt to address them with general-purpose methods, e.g. standard commercially available MILP solvers, would be too slow to address anything other than trivially small problem instances. The combination of such an approach with parallel computing would not provide a meaningful approach for real problems of commercial interest, since even accelerating this process by a factor of 100 is many times insufficient to compensate for the increased combinatorial complexity of practical applications. Any significant progress toward solving these problems must rest on the concept of understanding and exploiting problem structure. For this reason a significant portion of our effort in this project was devoted to building algorithms that take advantage of all available information concerning particular problem instances. Much of this information that gives the solver technology such an advantage comes from the fact that structural languages (DSPEC for design and RCSPEC for scheduling) describe the physical details of the problems in question in a much more

concentrated form than just a set of equations by themselves. Algorithm Engineering involves codifying this information and using it in the solver to guide underlying solutions based on physics in addition to strict numerical values, and is the key to solving problems of industrial scale.

All of the problems targeted in this project are essentially large scale MILPs of very special structure. Attempts to address these problems using general-purpose solvers have often failed, resulting in an accepted notion that mathematical programming techniques cannot succeed in this area. However, general-purpose solvers know very little about the problem they are solving. An MILP solver knows only the bounds on variables, which variables are restricted to take on integer values, and the expression of the objective function and constraints. Because process design problems are not just general MILPs, but MILPs resulting from a problem description expressed in a modeling language, the solution process can be aware of quite a bit about the underlying patterns of constraints and variables and how these relate to one-another. This information is readily available in the language, but would be very difficult to glean by examining the constraint matrix and variable bounds alone. A great deal of effort has been put forth in this project using this information as a guide for the solvers and the results have been impressive. The solver technology developed in this project has proven vastly superior to other products currently available, as evidenced by commercial success where other solutions have failed.

The algorithm engineering that performed in this project falls into three categories: customized linear algebra, reasoning on constraints, and pivot control within the LP solver. Customized linear algebraic methods allow dramatic speed enhancements inside the low-level workings of the MILP solver. This does not prune the search space, but does allow this space to be searched faster. Reasoning based on individual constraints allows better choices in exploring the solution space, and this does dramatically reduce the size of the space to be searched. This is a key component in solving industrial scale problems. The rationale is that these problems differ greatly from random or arbitrarily chosen MILPs. Scheduling and design problems contain structure that arises from the nature of their constraints. These constraints fall into a relatively small number of families and although they can interact in complex ways, these families produce patterns which tend to appear in many problems. For example, the dominant class of constraints in design and scheduling problems is those describing the material balance for a particular chemical constituent within a time bucket. These constraints require that the amount of material present at the end of the time span equals what was there at the beginning, plus what is produced or added, less what was consumed or taken away. There are a small number of such terms and they tend to interact in predictable ways with other constraint families. For example, the terms describing the amount of a material produced (on a piece of equipment) interact with constraints that limit the availability of that piece of equipment over time. If the method of production is a chemical reaction, then these terms will also interact with the material balance constraints for the chemicals that are feeds or products of that reaction. In a similar way, the interactions exist between variables. Consider the variables which describe the time and number of equipment items purchased in a design problem. The constraints describing the availability of these

items couple these variables in a simple way, namely if an item is purchased and installed in a time period, then that item will be available for use in all later time periods. This provides a simple coupling between these acquisition variables and we exploit this in our solver.

Pivot control is a special feature of the linear program (LP) solver implemented by APC. This LP solver forms the foundation for the MILP solver and serves as the basic low-level solution engine. Pivots are moves found by the LP solver that allow the value of the objective function to be improved and which maintain the validity of all constraints and variable bounds. Other LP solvers find and execute pivots based only on the problem constraints, variable bounds and objective function coefficients. Many times there are multiple choices for which pivot to execute that all appear numerically equivalent. Traditional solvers choose the pivot sequence without knowing anything about the types of variables and constraints affected by the available pivots. In this project, the ability to control the choice of pivots based on the type of variables they concern has been added . In addition, ways have been found to make these choices to enhance the probability of getting an integer feasible solution to the LP. Pivot control is also used in the search for integer feasible solutions that may not be cost optimal, but which, nevertheless, do satisfy all constraints.

The challenge of solving scheduling problems is to find a good quality solution that satisfies all constraints. Optimality is most often not necessary, and in fact may not be possible for in general for at least several more decades. What APC customers require in scheduling are high quality solutions that are feasible. From the customers' standpoint they need to solve the scheduling problem quickly and reliably and do it better than their competition. A quickly found solution which is not guaranteed to be optimal but which satisfies the constraints is of high commercial value. By contract, a solution that is optimal, but fails to satisfy material balance or other constraints is of much less value since it cannot be executed in reality, but must be adjusted (often an extremely labor intense process) to make it operable. For this reason, the use of a mathematical programming approach has proven critical to our technology. In an LP solver, the pivots represent moves from one solution to another, and these pivots describe only moves that satisfy constraints. That is, the pivots form the set of perturbations that connect feasible points in the search space. Any kind of randomized search would likely fail in chemical process problems due to the presence of a large number of strict material balance constraints (which are equality, not inequality constraints) that make it nearly impossible for a randomized perturbation to be feasible. The pivots provide a set of allowable perturbations, those that when executed may or may not improve the solution, but which will at least not violate constraints. For this reason, pivot control has been applied extensively in the scheduling solver.

Thus far, the techniques used to solve process configuration and scheduling problems have been discussed. As important as the base techniques are the methods for putting these techniques into practice. Pivot control is a good feature to have in an LP solver, but just how is this feature to be used to solve problems? To answer this question, the experimental process of iterative algorithm improvement was used, driven by real world

problems from APC customers. The simple description of this process is that a problem is encountered that the solver cannot solve, its performance is analyzed by tracing the execution and discover "root cause", the fundamental reason that causes the algorithm to fail on this problem. Once the root cause of the failure is understood, the solvers body of capability is evaluated to determine what can be done to alleviate the situation and how and where this correction should be implemented in the solver to grow its overall capability, in contrast to forking off several specialized solutions for each case, as is common practice in the industry.

Less sophisticated approaches that require a different custom algorithm for every application have at least two significant drawbacks. First, these algorithms have a narrow focus and are much more likely to fail when subjected to small perturbations in the problem structure. Second, there is the obvious problem of software maintenance. This has been the cause of the basic failure of artificial intelligence systems in addressing the complex area of process scheduling. What APC has done differently is that when an algorithm failure occurs, an algorithmic fix is provided or extensions so that the algorithm can solve the new problem, and can still solve every other problem that has previously been encountered. This makes the individual modifications much more difficult, but preserves the concept of a single algorithm overall, which is a much more attractive alternative from a business standpoint.

The greatest benefit produced by this approach is that the solver has much greater ability to extrapolate rather than just interpolate. In terms of addressing new problems in "out of the box" mode, the solver is far more robust than other commercial systems. For this reason, the solver is routinely used by engineering and process research and development groups at Fortune 500 companies to evaluate the operability of new designs before they are finalized. We have established this type of relationship with Eli Lilly, Proctor and Gamble, Coca-Cola and Searle Pharmaceuticals. These customers routinely use APC products as an integral part of their process development methodology. Sometimes, the algorithm fails in such constantly changing environments and must be analyzed to remedy this failure. Because of the concept of iterative improvement to the single algorithm, this action over the last few years has contributed greatly to the overall strength of the solver.

A number of features contribute to the difficulty of chemical process scheduling. Many software products on the market today do what is commonly called "infinite capacity scheduling", meaning they neglect such basic constraints as those which dictate two tasks cannot occur simultaneously on the same equipment. Obviously a scheduler that neglects these constraints is of little practical use in a chemical plant. The scheduler developed in this project treats the equipment allocation constraints rigorously and also handles constraints that dictate how close two tasks may approach each other on a particular piece of equipment (e.g. for clean-out reasons), time dependent availability of labor, utilities, or equipment, finite capacity storage for intermediates and so forth. Take for example, the common case where material is made in a tank, and then packed off into several different types of containers. Packing requires availability of both labor and free time on the packing equipment. Until the packing is completed, the process tank will be occupied

storing the remaining material for a variable amount of time. It is also typical of the chemical and pharmaceutical industries that these batch sizes are of a fixed size. This means that a sufficient need for material must be accumulated before an entire batch can be manufactured, and that storage must be available (often in the manufacturing vessel) to allow the entire batch to be worked off by subsequent tasks that require this material. While this situation is quite common in the process industries, it presents a unique and tight set of constraints that most commercial solvers are currently unable to handle.

There are other features of process scheduling that make it more difficult than other applications, e.g. parts assembly. Parts assembly scheduling is a convergent process, that is many parts come together to form fewer parts. In chemical plants however, it is not uncommon to have divergent processes, where many products are made from a few feedstocks. These problems are much more difficult to solve because many intermediates interact in a complex way with the supply of a large number of final products. Storage of intermediates is another complicating factor since chemicals, in contrast to discrete parts, cannot be mixed in a storage tank and cannot be set aside on a pallet. Therefore, getting the timing absolutely correct in coordinating the production and consumption of intermediates is also very important in chemical and pharmaceutical scheduling.

Customers have come to rely on APC to maintain a lead in technology by continually extending the capabilities of the solver engine to include new larger envelopes of performance. We have developed special relationships with the above mentioned large companies' engineering and process development organizations. APC provides the best available process scheduling and design technology and rapid response to scheduling difficulties. In return, APC's algorithm development staff receives a constant supply of challenging real world problems that stress the solution engine and allow it to continually improve.

Example Cases

In order to demonstrate the scale and complexity of the problems addressed by techniques developed in this project, several modified industrial examples are presented in this section. A design example will be followed by two schedules.

APC was presented with a design situation involving a business unit with 14 main products, some of which were currently under production, and others of which were planned for future production but were not currently being manufactured. A set of geographically distributed manufacturing facilities was in existence and was actively involved in producing a subset of the overall product mix. Each product involved several stages of production. The overriding question was where to allocate products for production, plan new capacity at existing facilities, plan the construction of new facilities and warehouses, and assess the advantages possible in using contract manufacturers. A variety of factors have influence over the solution including equipment capital and operating costs, location dependent tax rates, and alternate process technologies. The basic production flow of one of the products is shown in Figure 1.

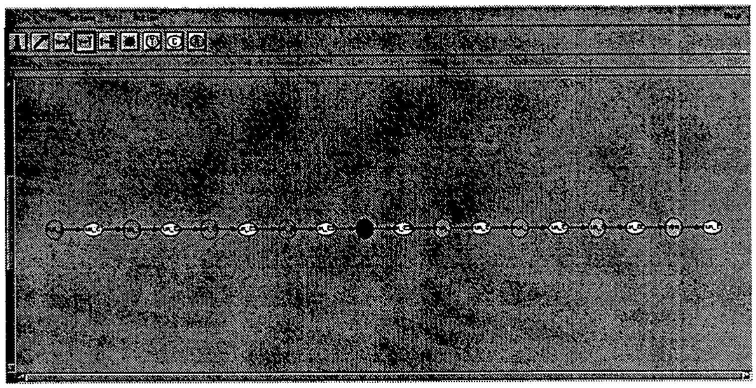

Figure 1 – An Existing Product

Two separate geographic regions are used, as indicated by the different shades of production tasks shown on the two sides of the diagram. As currently produced with current manufacturing assets in place, this diagram describes the flow of materials from left to right being transformed from one intermediate to another, and finally into finished product. The dark oval in the middle marks the separation of geographic regions. To this current configuration, many possible changes can be made, as more fully depicted in Figure 2.

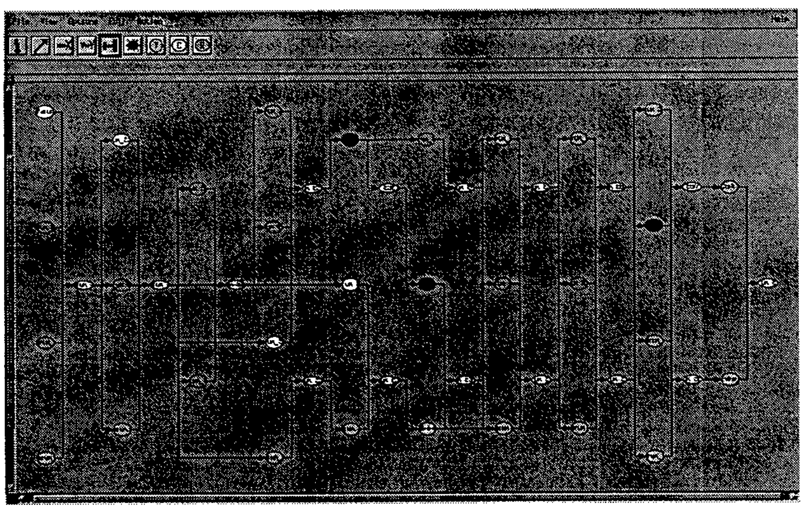

Figure 2 – Candidate Configuration Options

In this figure, each stage from the previous diagram has now been split into multiple parallel ovals. The significance of this is that one of the ovals in each stage of Figure 2 corresponds to an oval in Figure 1. Each other oval in Figure 2 represents an option that does not currently exist physically. These options include adding additional capacity at existing sites, re-locating production from existing sites to new sites, and using a contractor. The design optimizer then solves a problem that includes all of these options and their respective costs to produce a maximal Net Present Value plan for building and reorganizing manufacturing assets well out into the future. The example shown includes only 1 of the 14 products, and therefore, the full problem is quite complex.

Subsequent downstream distribution of materials produced in examples like those shown above often result in problems where warehousing decisions and transportation network configurations must be made, as shown in Figure 3.

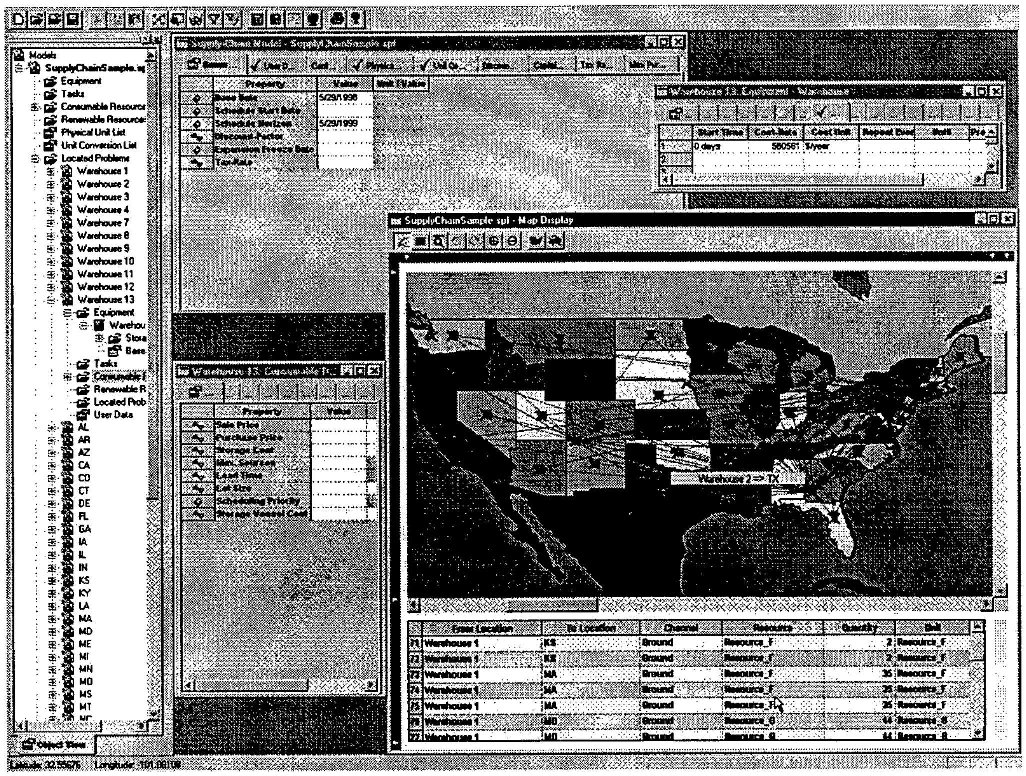

Figure 3 – Distribution Network

As discussed earlier, within each configured production facility, the second component of the problem is making sure that the configuration is operable from a detailed scheduling standpoint. The diagrams in Figures 4-7 show the material flow diagrams and corresponding schedules in two such large applications. In each case, the production chain involves many steps, and many end products. Each schedule requires placement of thousands of tasks subject to finite resources. In both cases, the engineered algorithms developed in this project required only minutes of computer time, whereas these problems have proven intractable for other commercial scheduling technology.

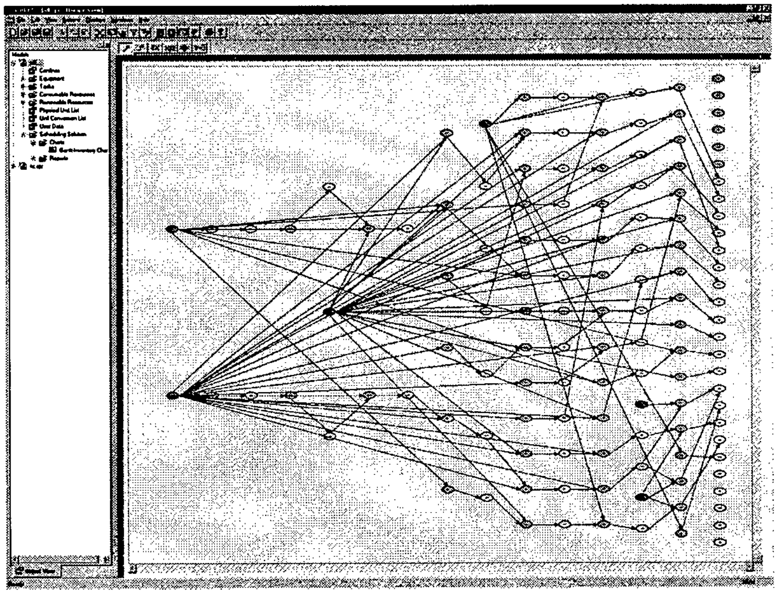

Figure 4 – Complex Process Flow 1

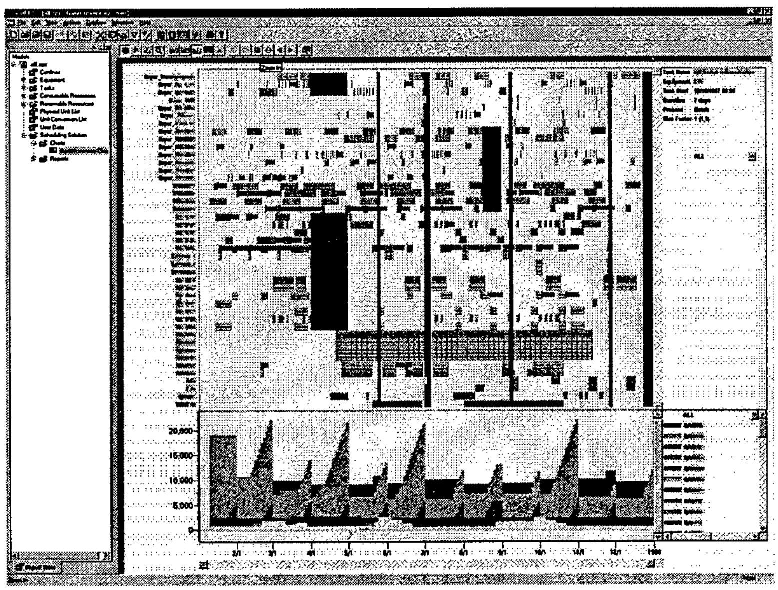

Figure 5 – Large Schedule 1

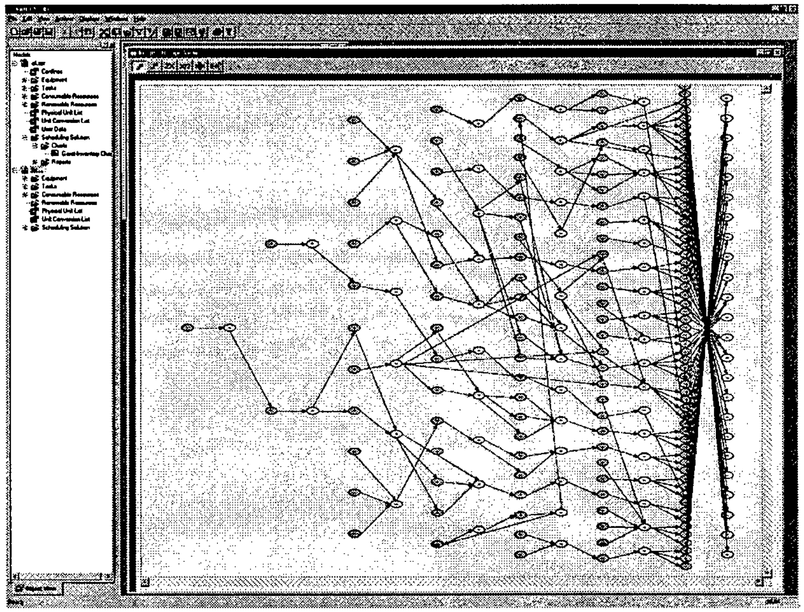

Figure 6 – Complex Process Flow 2

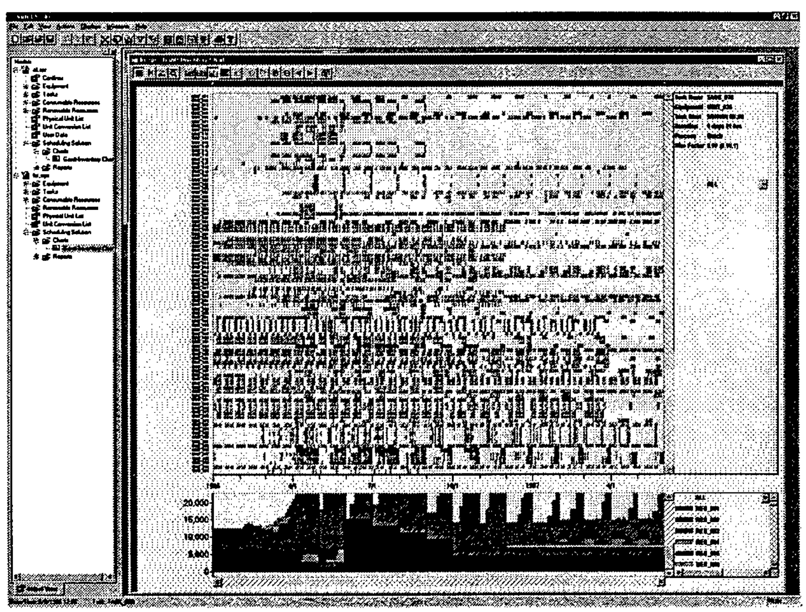

Figure 7 – Large Schedule 2

Revenue Growth

Since the first year of its operation in 1994, APC has grown from a startup company relying heavily on small business grants to a self-reliant business with rapidly expanding revenues and payroll. Revenues grew from $85k in 1994 to $465k in 1997. However, a

significant portion of the revenue in 1997 came from research grants, the principal contributor to this revenue being this contract, DAH DAAH04-96-C-0015. The technology APC developed over the life of this contract has proven to be a great commercial success and is the major contributor to our current sales revenue. With only commercial funding, APC would not have been able to perform the basic research and development necessary to bring these new ideas to the marketplace. In 1998, APC revenues were at $901k, with very little coming from research grants. In the first two quarters of 1999, revenues were $665k and are on track to achieve between $1.5 and 1.9 million in revenues for this year. Demonstrating the success of the research funding we received, none of our 1999 revenues will come from research grants.

Current revenues are not only coming from Tier 1 companies like those mentioned above directly applying this technology, but also new business areas that are emerging and that can benefit from the same advanced technology developed in this project. These areas include resource planning for product and process research and development, where a more advanced method of incorporating uncertainty becomes important, warehouse management and optimization, where the time scale of operations reduces from hours to seconds, and third party technology sublicensing, where other software systems that can benefit from advanced optimization technology can utilize many of the subroutines used in APC's own applications.

Conclusions

In summary, the research performed in this project included core optimization system theory and design, implementation of that technology in the practical arena of manufacturing process design, and subsequent delivery of this technology in a commercial software system. This work has contributed strongly to laying the foundations for a successful business. APC has achieved a level of business where it has overcome the technological hurdles often associated with a high-tech business. The next major challenge will be from a business standpoint to answer the challenge of taking very strong technology and successfully marketing it to a much wider audience than APC's current clientele.